In [1]:

```
#Importing Libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

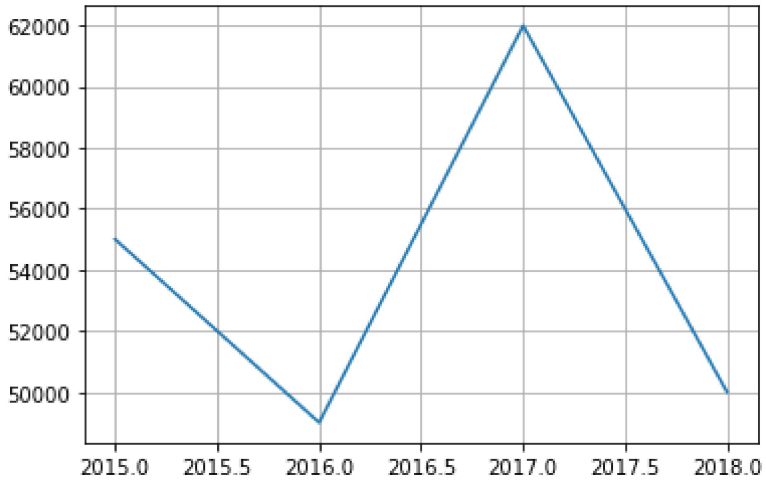
2D- Line Plot

In [3]:

```
# Bivariate Analysis
# For Categorical-Numerical and Numerical- Numerical,
# It is plot on Time series data.
```

In [2]:

```
#ploting a simple graph
price= [55000,49000,62000,50000]
year= [2015,2016,2017,2018]

#plt.plot(x-axis,y-axis)
plt.plot(year,price)
plt.grid()
plt.show()
```



In [7]:

```
#Plotting by loading a dataset
Batsman= pd.read_csv("sharma-kohli.csv")
Batsman.head()
```

Out[7]:

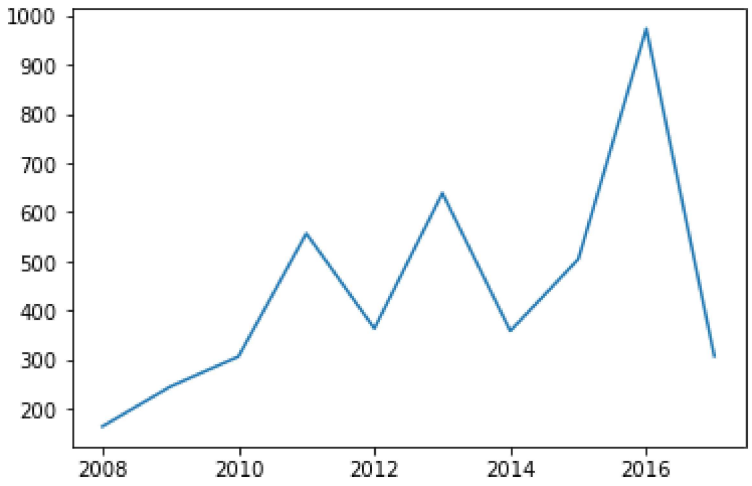
	index	RG Sharma	V Kohli
0	2008	404	165
1	2009	362	246
2	2010	404	307
3	2011	372	557
4	2012	433	364

In [10]:

```
plt.plot(Batsman["index"],Batsman["V Kohli"])
```

Out[10]:

```
[<matplotlib.lines.Line2D at 0x22d77b5f550>]
```

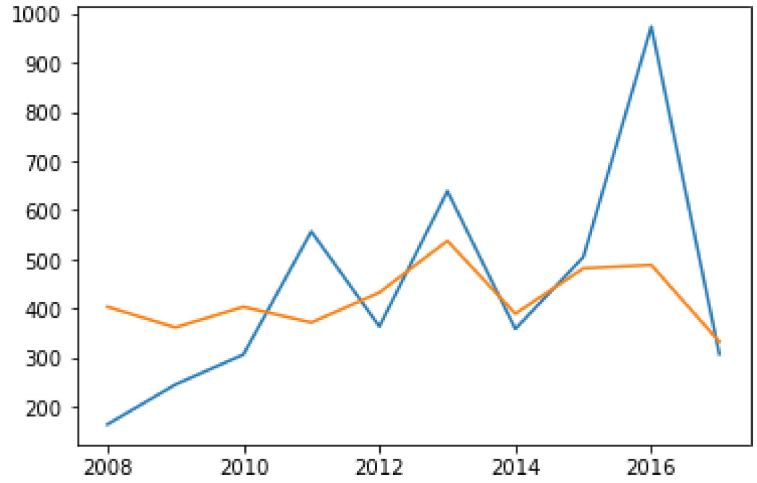


In [11]:

```
#Ploting multiple graphs
plt.plot(Batsman["index"],Batsman["V Kohli"])
plt.plot(Batsman["index"],Batsman["RG Sharma"])
```

Out[11]:

```
[<matplotlib.lines.Line2D at 0x22d77af9e80>]
```

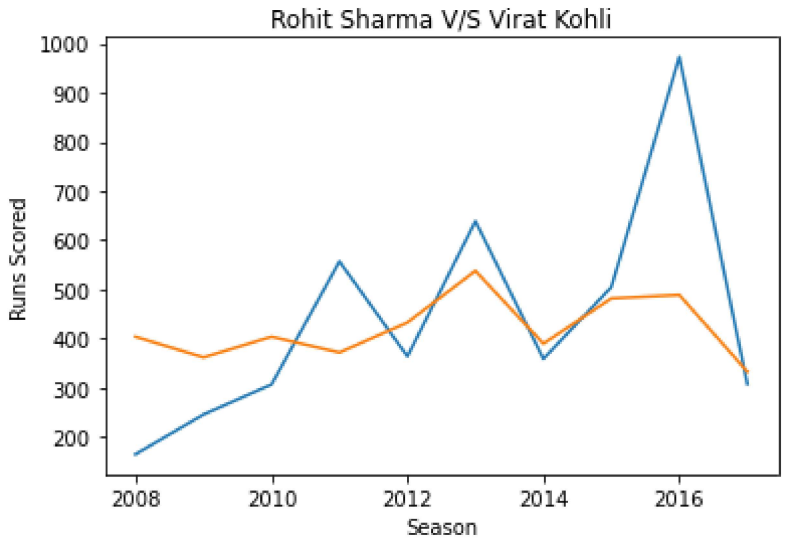


In [14]:

```
#Ploting with title

plt.plot(Batsman["index"],Batsman["V Kohli"])
plt.plot(Batsman["index"],Batsman["RG Sharma"])

plt.title("Rohit Sharma V/S Virat Kohli")
plt.xlabel("Season")
plt.ylabel("Runs Scored")
```

Out[14]:

```
Text(0, 0.5, 'Runs Scored')
```

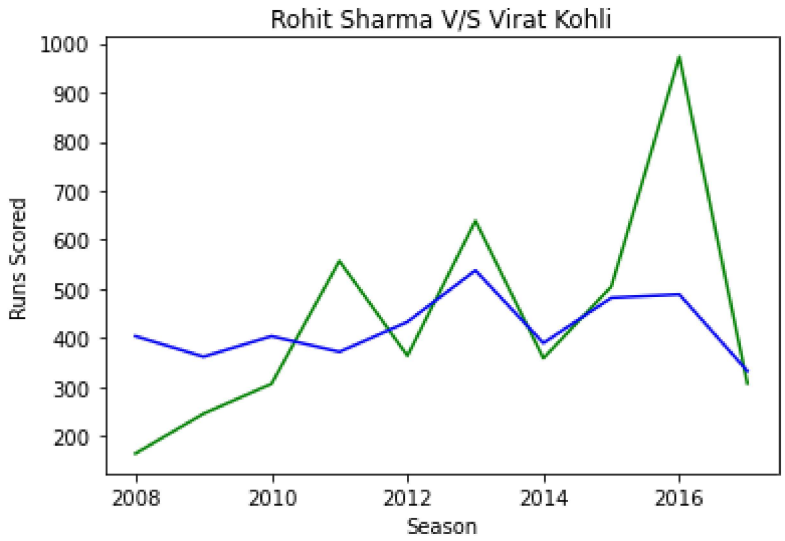


In [15]:

```
# Changing the colour (hex color)

plt.plot(Batsman["index"],Batsman["V Kohli"], color="green")
plt.plot(Batsman["index"],Batsman["RG Sharma"], color="blue")

plt.title("Rohit Sharma V/S Virat Kohli")
plt.xlabel("Season")
plt.ylabel("Runs Scored")
```

Out[15]:

```
Text(0, 0.5, 'Runs Scored')
```

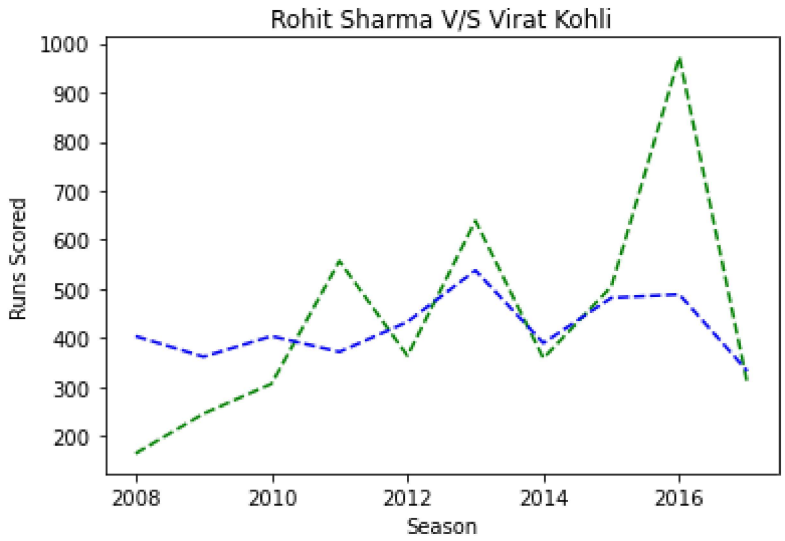


In [16]:

```
# Other than solid line(dashed, dotted, dashdot)

plt.plot(Batsman["index"],Batsman["V Kohli"], color="green", linestyle="dashed")
plt.plot(Batsman["index"],Batsman["RG Sharma"], color="blue", linestyle="dashed")

plt.title("Rohit Sharma V/S Virat Kohli")
plt.xlabel("Season")
plt.ylabel("Runs Scored")
```

Out[16]:

```
Text(0, 0.5, 'Runs Scored')
```

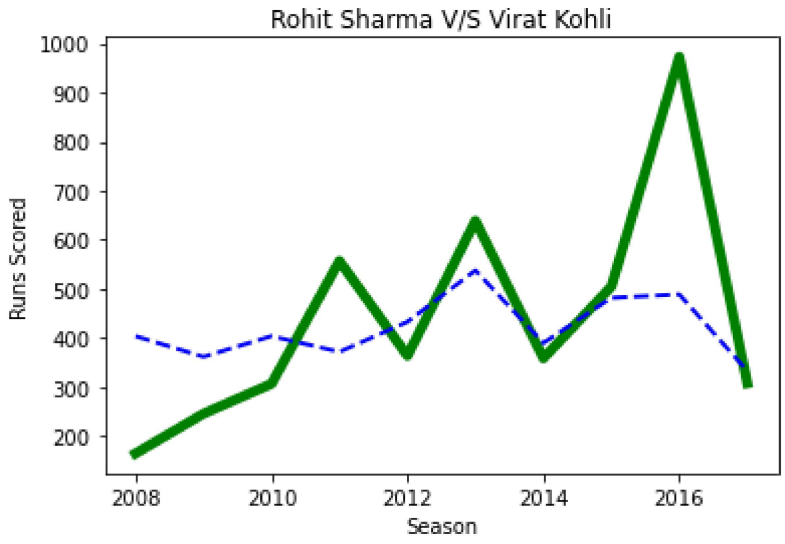


In [20]:

```
# Line width

plt.plot(Batsman["index"],Batsman["V Kohli"], color="green", linestyle="solid", linewidt
plt.plot(Batsman["index"],Batsman["RG Sharma"], color="blue", linestyle="dashed", linewi

plt.title("Rohit Sharma V/S Virat Kohli")
plt.xlabel("Season")
plt.ylabel("Runs Scored")
```

Out[20]:

```
Text(0, 0.5, 'Runs Scored')
```

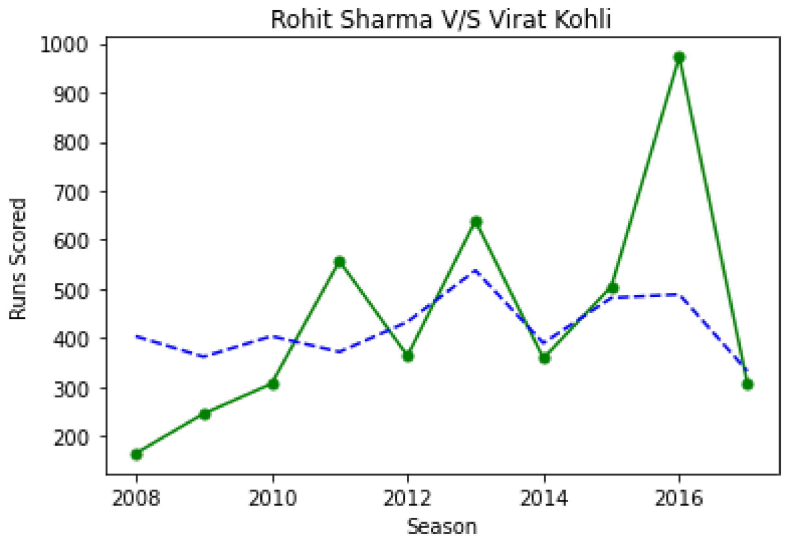


In [28]:

```
# Marker

plt.plot(Batsman["index"],Batsman["V Kohli"], color="green", marker=".", markersize=10 )
plt.plot(Batsman["index"],Batsman["RG Sharma"], color="blue", linestyle="dashed")

plt.title("Rohit Sharma V/S Virat Kohli")
plt.xlabel("Season")
plt.ylabel("Runs Scored")
```

Out[28]:

```
Text(0, 0.5, 'Runs Scored')
```

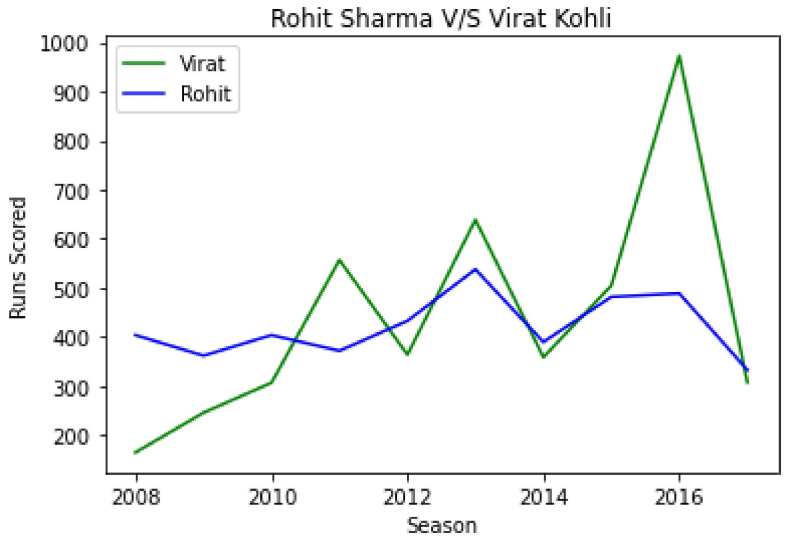


In [30]:

```
# Using Lagend

plt.plot(Batsman["index"],Batsman["V Kohli"], color="green", label="Virat")
plt.plot(Batsman["index"],Batsman["RG Sharma"], color="blue",label="Rohit")

plt.title("Rohit Sharma V/S Virat Kohli")
plt.xlabel("Season")
plt.ylabel("Runs Scored")

plt.legend()
```

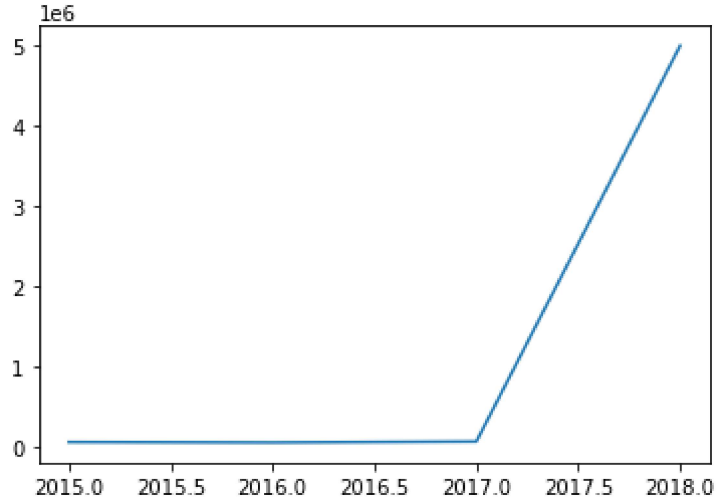
Out[30]:

```
<matplotlib.legend.Legend at 0x22d79e3c9d0>
```



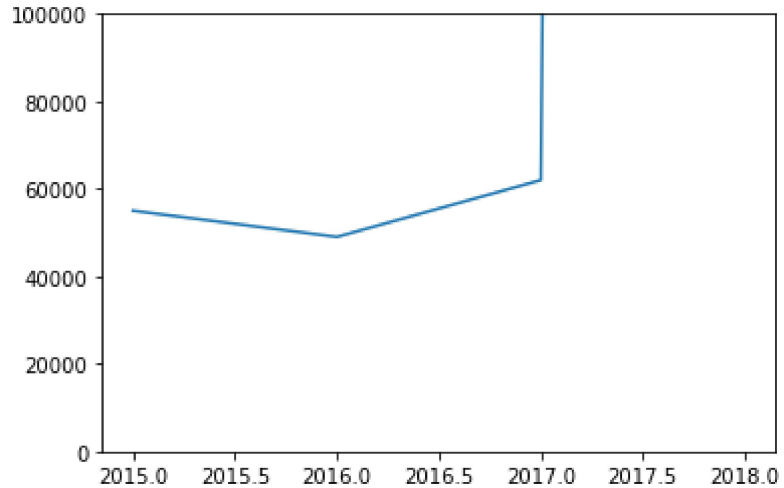
In [31]:

```
# Limiting Axes (Mainly used for outliers)
#(Bcoz of outliers graps can get flat or not accurate, we can trim for this type)


price= [55000,49000,62000,5000000]
year= [2015,2016,2017,2018]

#plt.plot(x-axis,y-axis)
plt.plot(year,price)
plt.show()
```



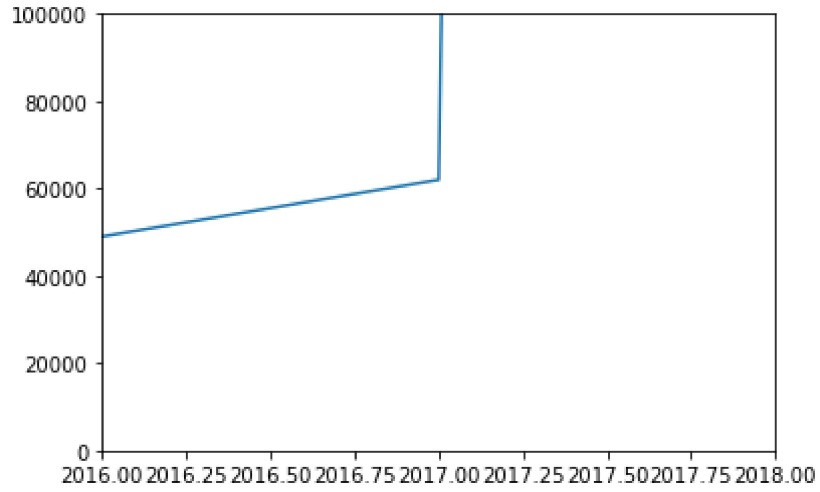
In [32]:

```
price= [55000,49000,62000,5000000]
year= [2015,2016,2017,2018]

#plt.plot(x-axis,y-axis)
plt.plot(year,price)
plt.ylim(0,100000)
plt.show()
```



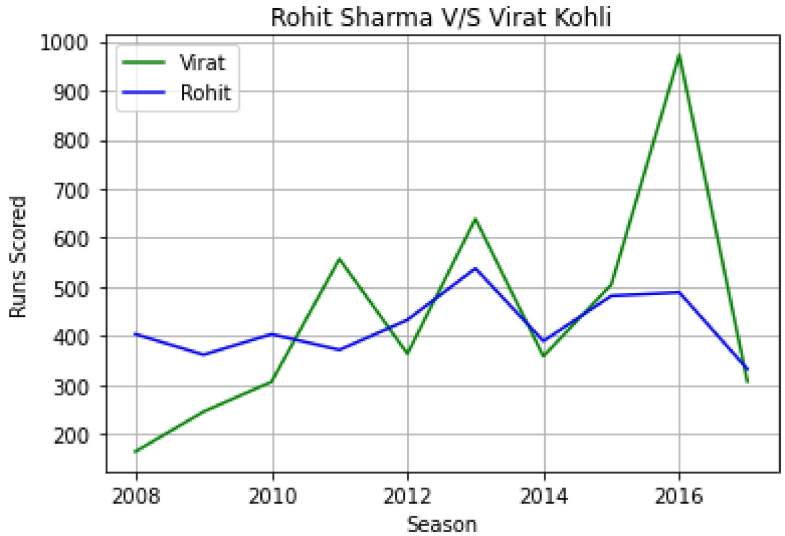
In [33]:

```
price= [55000,49000,62000,5000000]
year= [2015,2016,2017,2018]

#plt.plot(x-axis,y-axis)
plt.plot(year,price)
plt.ylim(0,100000)
plt.xlim(2016,2018)
plt.show()
```



In [34]:

```
# Grid


plt.plot(Batsman["index"],Batsman["V Kohli"], color="green", label="Virat")
plt.plot(Batsman["index"],Batsman["RG Sharma"], color="blue",label="Rohit")

plt.title("Rohit Sharma V/S Virat Kohli")
plt.xlabel("Season")
plt.ylabel("Runs Scored")

plt.legend()
plt.grid()
```



Scatter Plot

In [3]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:

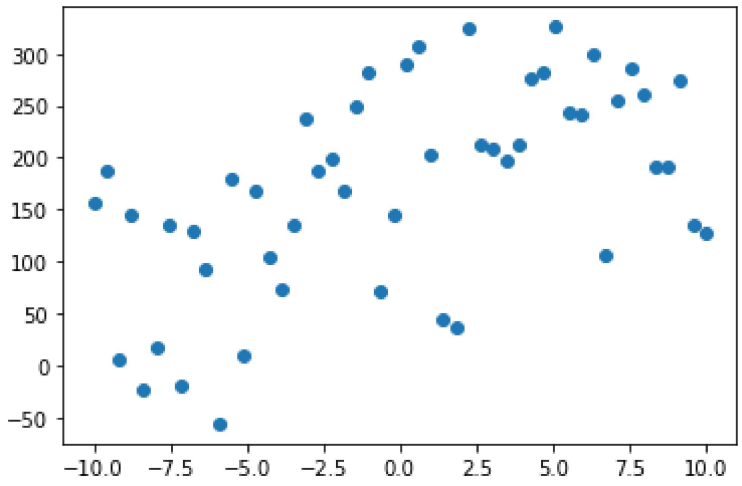
```
# Bivariate Analysis
# Numerical-Numerical
# Finding Correlation between two quantities
# (2d plot is the main output of scatter plot)
```

In [5]:

```
# Plotting simple scatter plot
x= np.linspace(-10,10,50)
x
y= 10*x +3+np.random.randint(0,300,50)
y

plt.scatter(x,y)
```

Out[5]:

```
<matplotlib.collections.PathCollection at 0x18aa7228e20>
```



In [6]:

```
#Scatter plot on pandas dataframe
Data= pd.read_csv("batter.csv")
df= Data.head(50)
df
```

Out[6]:

	batter	runs	avg	strike_rate
0	V Kohli	6634	36.251366	125.977972
1	S Dhawan	6244	34.882682	122.840842
2	DA Warner	5883	41.429577	136.401577
3	RG Sharma	5881	30.314433	126.964594
4	SK Raina	5536	32.374269	132.535312
5	AB de Villiers	5181	39.853846	148.580442
6	CH Gayle	4997	39.658730	142.121729
7	MS Dhoni	4978	39.196850	130.931089
8	RV Uthappa	4954	27.522222	126.152279
9	KD Karthik	4377	26.852761	129.267572
10	G Gambhir	4217	31.007353	119.665153
11	AT Rayudu	4190	28.896552	124.148148
12	AM Rahane	4074	30.863636	117.575758
13	KL Rahul	3895	46.927711	132.799182
14	SR Watson	3880	30.793651	134.163209
15	MK Pandey	3657	29.731707	117.739858
16	SV Samson	3526	29.140496	132.407060
17	KA Pollard	3437	28.404959	140.457703
18	F du Plessis	3403	34.373737	127.167414
19	YK Pathan	3222	29.290909	138.046272
20	BB McCullum	2882	27.711538	126.848592
21	RR Pant	2851	34.768293	142.550000
22	PA Patel	2848	22.603175	116.625717
23	JC Buttler	2832	39.333333	144.859335
24	SS Iyer	2780	31.235955	121.132898
25	Q de Kock	2767	31.804598	130.951254
26	Yuvraj Singh	2754	24.810811	124.784776
27	V Sehwag	2728	27.555556	148.827059
28	SA Yadav	2644	29.707865	134.009123
29	M Vijay	2619	25.930693	118.614130
30	RA Jadeja	2502	26.617021	122.108346
31	SPD Smith	2495	34.652778	124.812406
32	SE Marsh	2489	39.507937	130.109775
33	DA Miller	2455	36.102941	133.569097
34	JH Kallis	2427	28.552941	105.936272
35	WP Saha	2427	25.281250	124.397745
36	DR Smith	2385	28.392857	132.279534

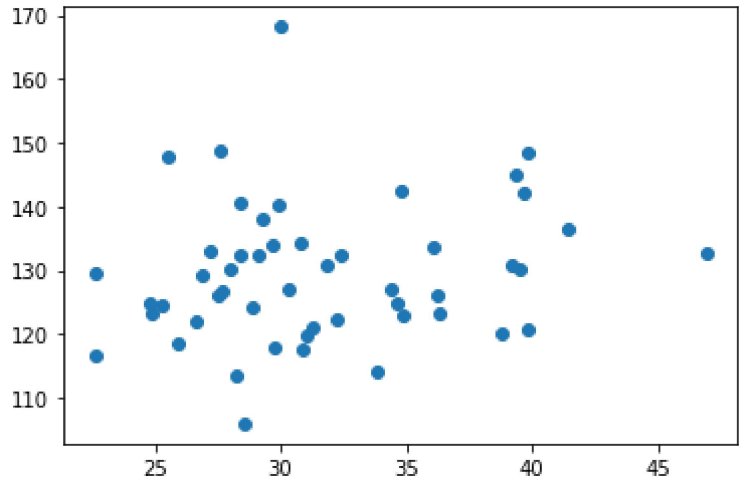
	batter	runs	avg	strike_rate
37	MA Agarwal	2335	22.669903	129.506378
38	SR Tendulkar	2334	33.826087	114.187867
39	GJ Maxwell	2320	25.494505	147.676639
40	N Rana	2181	27.961538	130.053667
41	R Dravid	2174	28.233766	113.347237
42	KS Williamson	2105	36.293103	123.315759
43	AJ Finch	2092	24.904762	123.349057
44	AC Gilchrist	2069	27.223684	133.054662
45	AD Russell	2039	29.985294	168.234323
46	JP Duminy	2029	39.784314	120.773810
47	MEK Hussey	1977	38.764706	119.963592
48	HH Pandya	1972	29.878788	140.256046
49	Shubman Gill	1900	32.203390	122.186495

In [7]:

```
plt.scatter(df["avg"],df["strike_rate"])
```

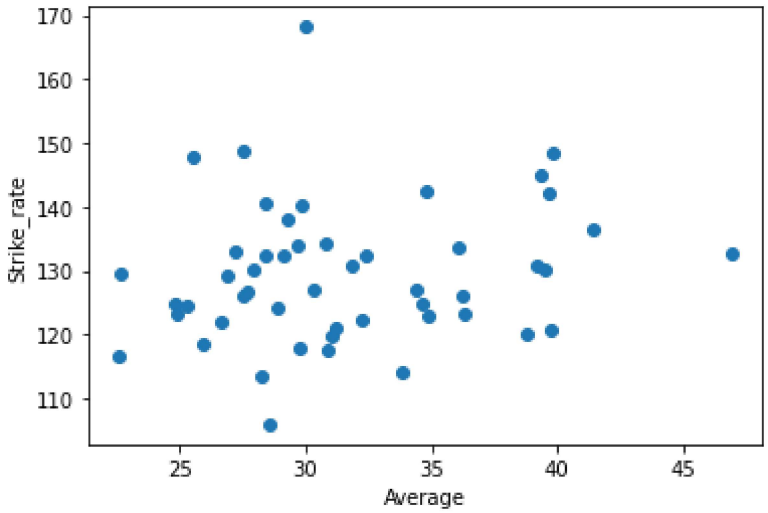
Out[7]:

```
<matplotlib.collections.PathCollection at 0x18aa794d970>
```



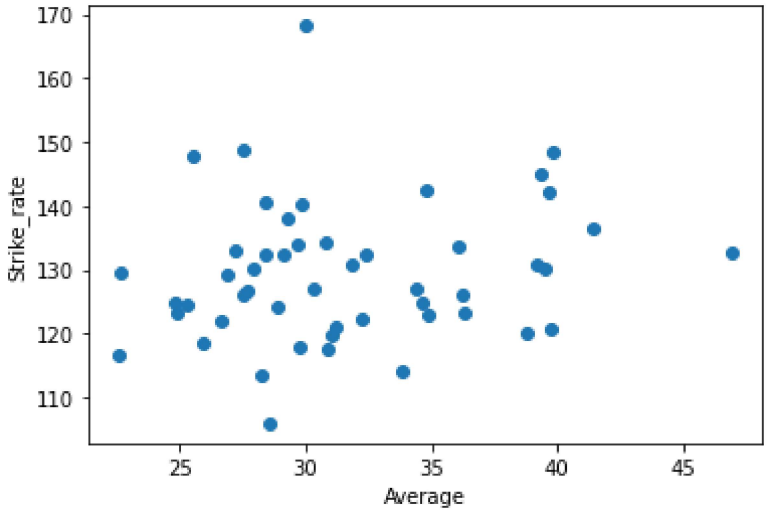
In [8]:

```
# Labeling
plt.scatter(df["avg"],df["strike_rate"])
plt.xlabel("Average")
plt.ylabel("Strike_rate")

plt.show()
```



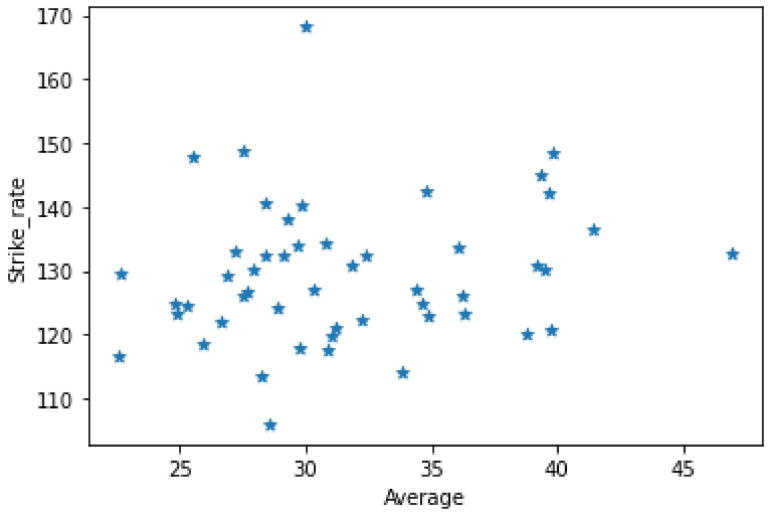
In [9]:

```
# Labeling
plt.scatter(df["avg"],df["strike_rate"])
plt.xlabel("Average")
plt.ylabel("Strike_rate")

plt.show()
```



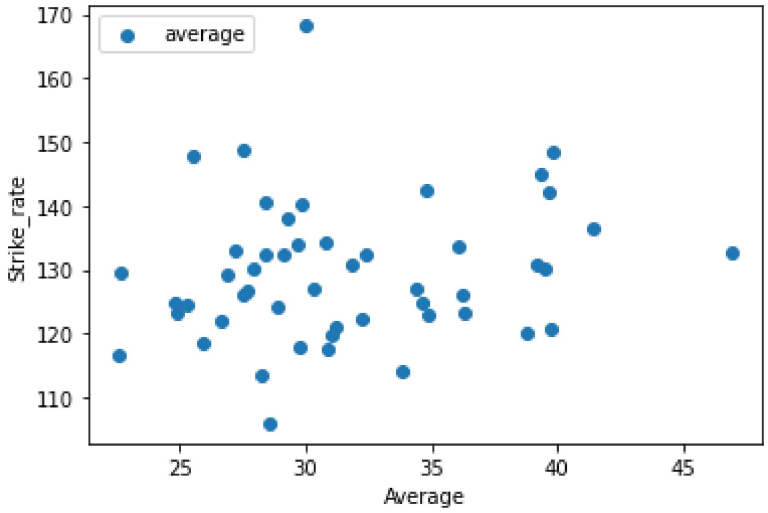
In [10]:

```
#Using Marker
plt.scatter(df["avg"],df["strike_rate"], marker="*")
plt.xlabel("Average")
plt.ylabel("Strike_rate")

plt.show()
```



In [11]:

```
# Legend- Location od Legend
plt.scatter(df["avg"],df["strike_rate"], label="average")
plt.xlabel("Average")
plt.ylabel("Strike_rate")

plt.legend(loc="upper left")
plt.show()
```



In [12]:

```
#Size- LOADING A DATASET

tips= sns.load_dataset("tips")
tips
```

Out[12]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

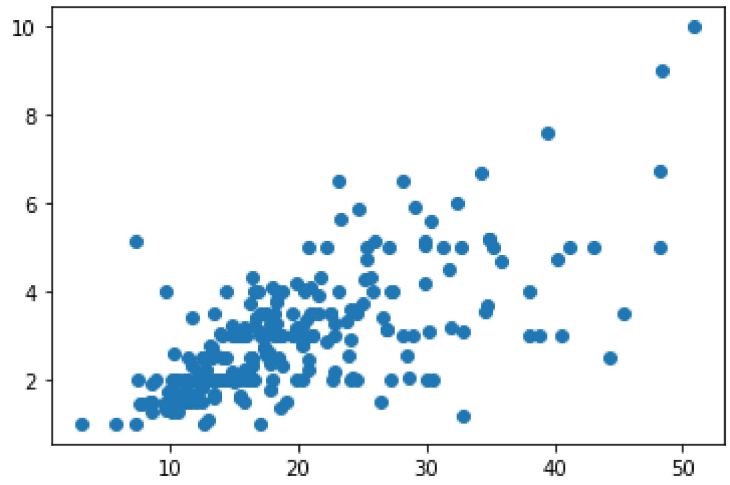
244 rows × 7 columns

In [13]:

```
plt.scatter(tips["total_bill"],tips["tip"])
```

Out[13]:

```
<matplotlib.collections.PathCollection at 0x18aa7c4de50>
```

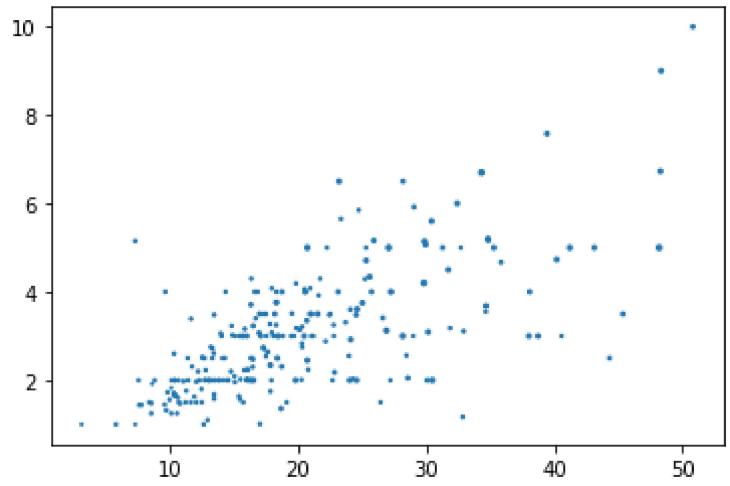


In [14]:

```
# Size will give us the number of person the customer came with
plt.scatter(tips["total_bill"],tips["tip"], s=tips["size"])
```

Out[14]:

```
<matplotlib.collections.PathCollection at 0x18aa7cb6610>
```

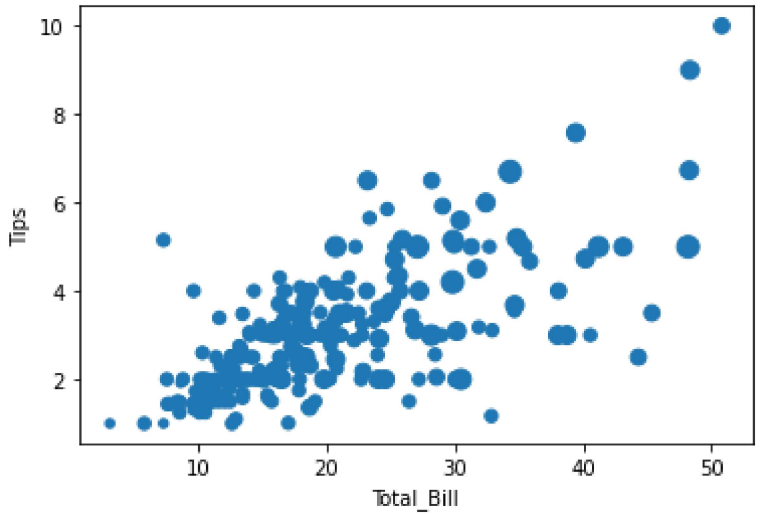


In [15]:

```
# Multiplying it by 20 to increase the size of the plots

plt.scatter(tips["total_bill"],tips["tip"], s=tips["size"]*20)
plt.xlabel("Total_Bill")
plt.ylabel("Tips")
```

Out[15]:

```
Text(0, 0.5, 'Tips')
```

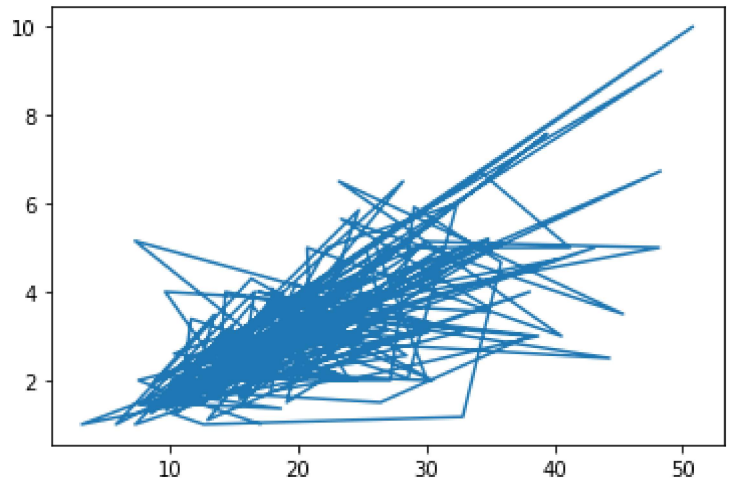


In [16]:

```
plt.plot(tips["total_bill"],tips["tip"])
```

Out[16]:

```
[<matplotlib.lines.Line2D at 0x18aa7bb3b50>]
```

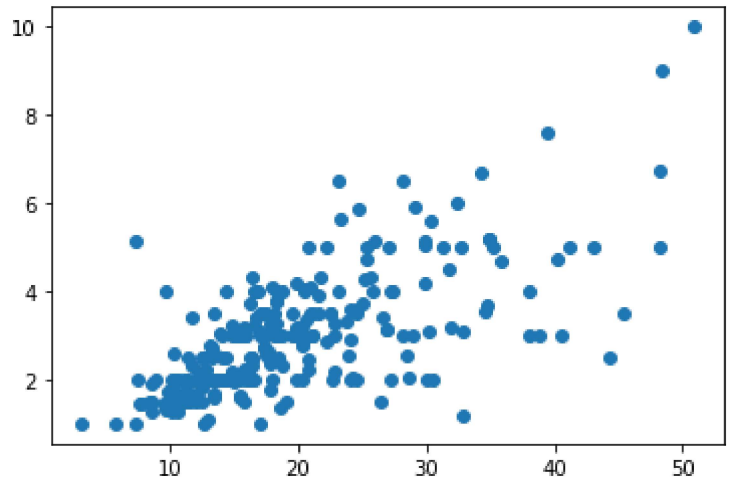


In [17]:

```
#Faster technique to plot scatter plot
plt.plot(tips["total_bill"],tips["tip"],"o" )
```

Out[17]:

```
[<matplotlib.lines.Line2D at 0x18aa7b975e0>]
```



In [18]:

```
# Note- we should not use plt.plot with "o" bcoz there are many parameters we can not us
```

Bar Plot

In [19]:

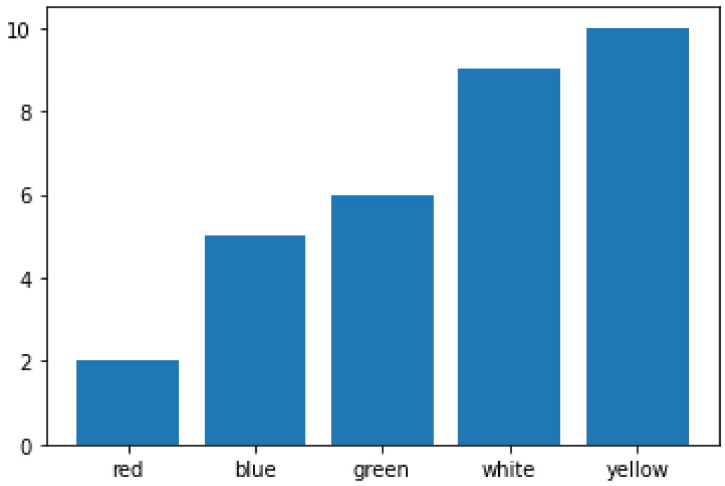
```
#x axis- categories
#y axis- Numerical
#Bivariate analysis
# Use case- Aggregate analysis of groups
# Numerical vs Categorical
```

In [4]:

```
# Simple bar chart

children=[2,5,6,9,10]
color=["red","blue","green","white","yellow"]

plt.bar(color,children)
```

Out[4]:

```
<BarContainer object of 5 artists>
```

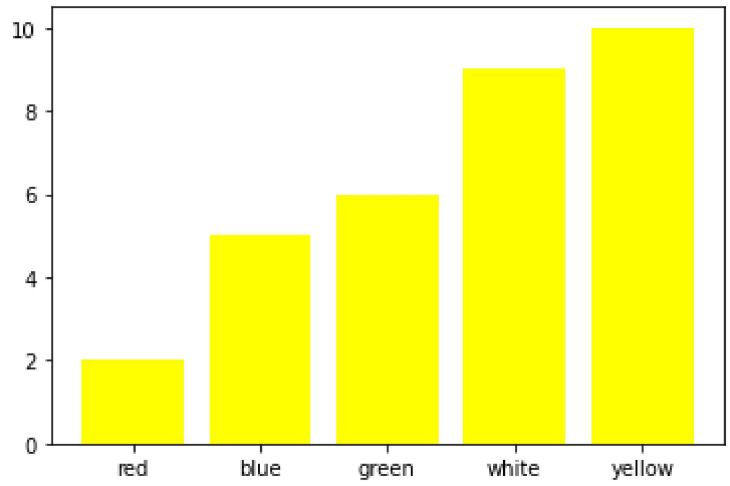


In [5]:

```
# Color

children=[2,5,6,9,10]
color=["red","blue","green","white","yellow"]

plt.bar(color,children, color="Yellow")
```

Out[5]:

```
<BarContainer object of 5 artists>
```

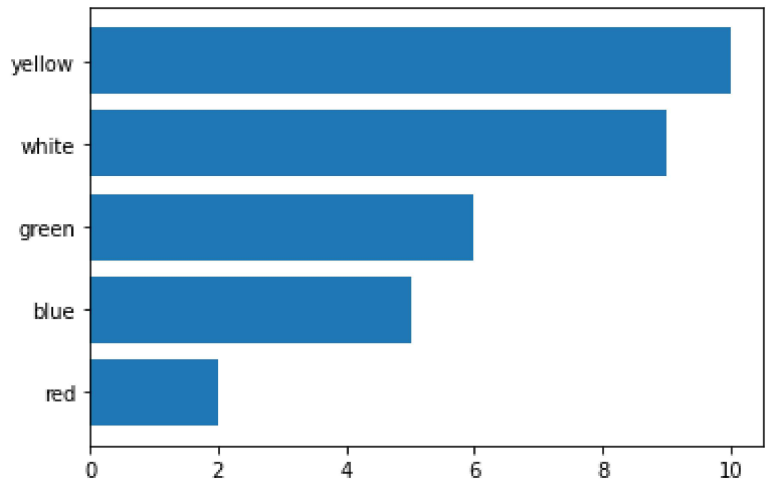


In [6]:

```
# Horizontal bar chart (Barh)

children=[2,5,6,9,10]
color=["red","blue","green","white","yellow"]

plt.barh(color,children)
```

Out[6]:

```
<BarContainer object of 5 artists>
```



In [7]:

```
# Colour

df= pd.read_csv("batsman_season_record.csv")
df
```

Out[7]:

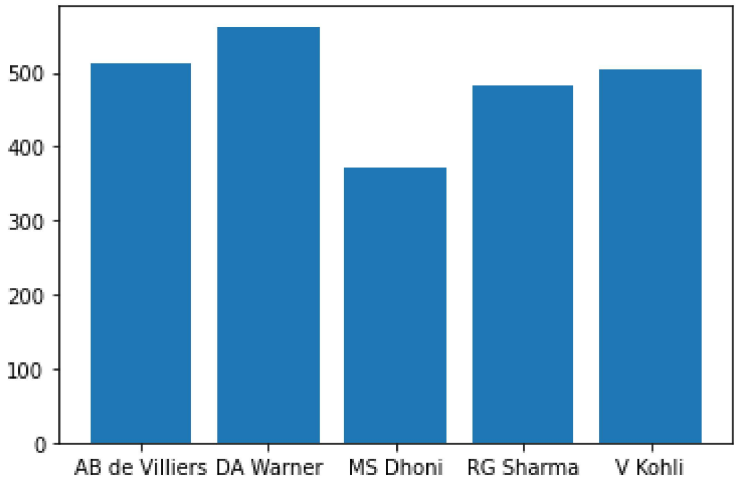
	batsman	2015	2016	2017
0	AB de Villiers	513	687	216
1	DA Warner	562	848	641
2	MS Dhoni	372	284	290
3	RG Sharma	482	489	333
4	V Kohli	505	973	308

In [9]:

```
plt.bar(df["batsman"],df["2015"])
```

Out[9]:

```
<BarContainer object of 5 artists>
```

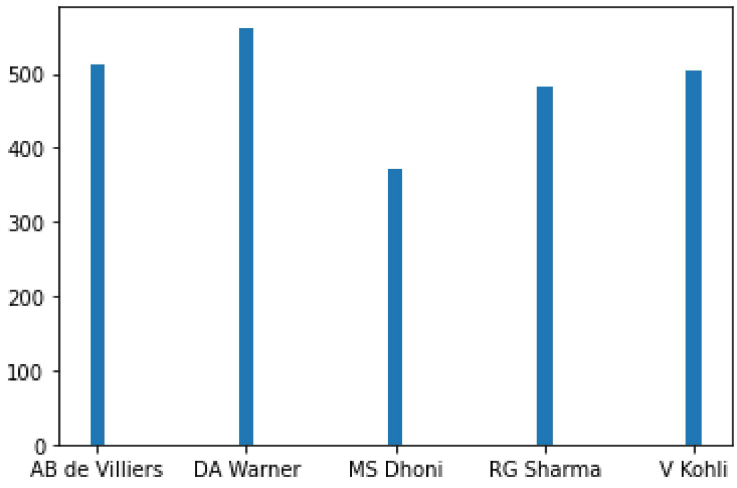


In [11]:

```
#Plot multiple graph

plt.bar(df["batsman"],df["2015"], width=0.1)
```

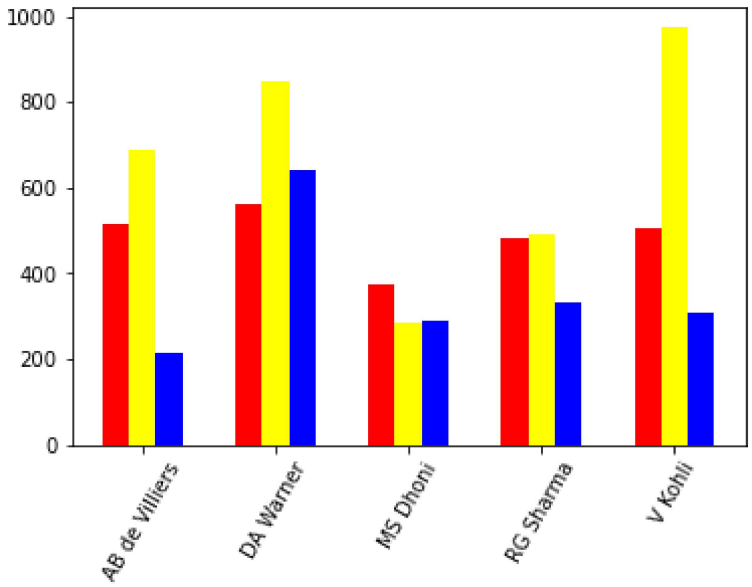
Out[11]:

```
<BarContainer object of 5 artists>
```



In [24]:

```
plt.bar(np.arange(df.shape[0])-0.2,df["2015"], width=0.2,color="Red")
plt.bar(np.arange(df.shape[0]),df["2016"], width=0.2,color="Yellow")
plt.bar(np.arange(df.shape[0])+0.2,df["2017"], width=0.2,color="Blue")

plt.xticks(np.arange(df.shape[0]), df["batsman"], rotation=60)

plt.show()
```



In [25]:

```
# Stacked bar chart
df
```

Out[25]:

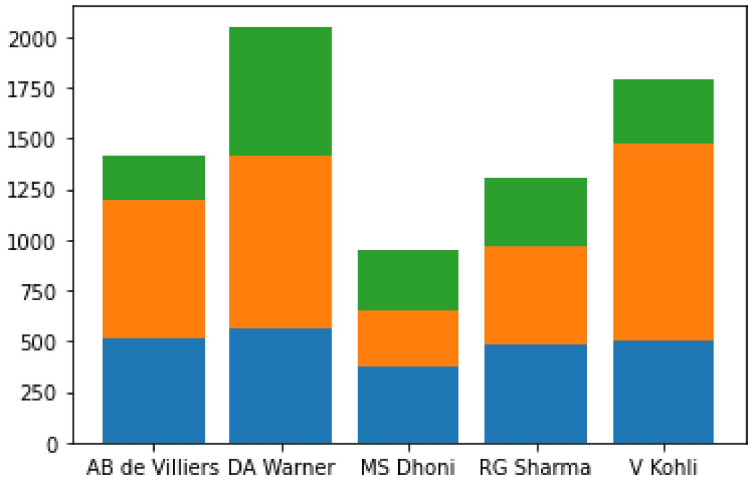
	batsman	2015	2016	2017
0	AB de Villiers	513	687	216
1	DA Warner	562	848	641
2	MS Dhoni	372	284	290
3	RG Sharma	482	489	333
4	V Kohli	505	973	308

In [29]:

```
plt.bar(df["batsman"],df["2015"])
plt.bar(df["batsman"],df["2016"], bottom=df["2015"])
plt.bar(df["batsman"],df["2017"], bottom=df["2016"]+ df["2015"])
```

Out[29]:

```
<BarContainer object of 5 artists>
```

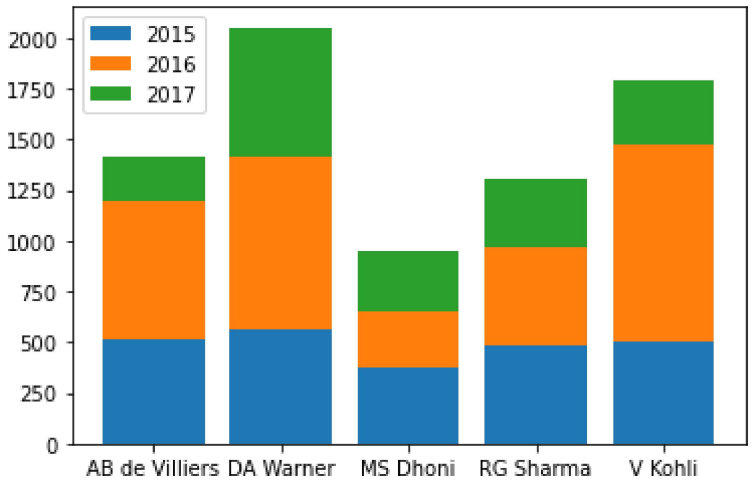


In [30]:

```
plt.bar(df["batsman"],df["2015"],label="2015")
plt.bar(df["batsman"],df["2016"], bottom=df["2015"],label="2016")
plt.bar(df["batsman"],df["2017"], bottom=df["2016"]+ df["2015"],label="2017")

plt.legend()
plt.show
```

Out[30]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Histogram

In [32]:

```
# Univarate Analysis
# Numerical Columns
# Use case- Frequency count
```

In [20]:

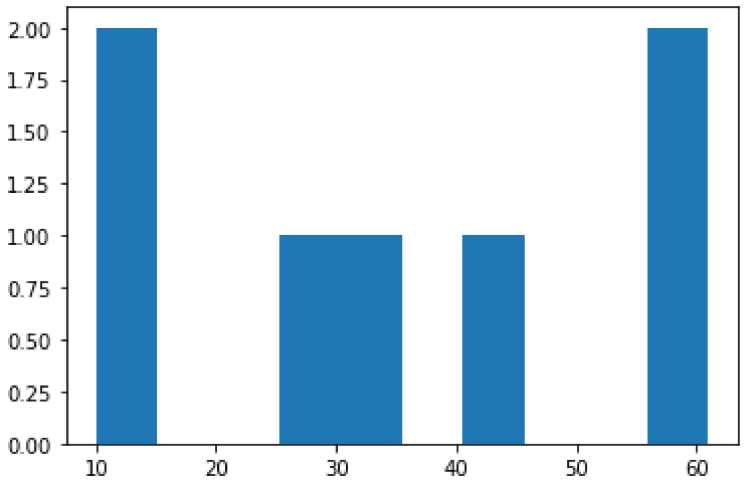
```
# Creatings bins
```

In [21]:

```
#simple data

data=[32,45,56,10,15,27,61]
plt.hist(data)
```

Out[21]:

```
(array([2., 0., 0., 1., 1., 0., 1., 0., 0., 2.]),
 array([10. , 15.1, 20.2, 25.3, 30.4, 35.5, 40.6, 45.7, 50.8, 55.9, 61.
]),
 <BarContainer object of 10 artists>)
```

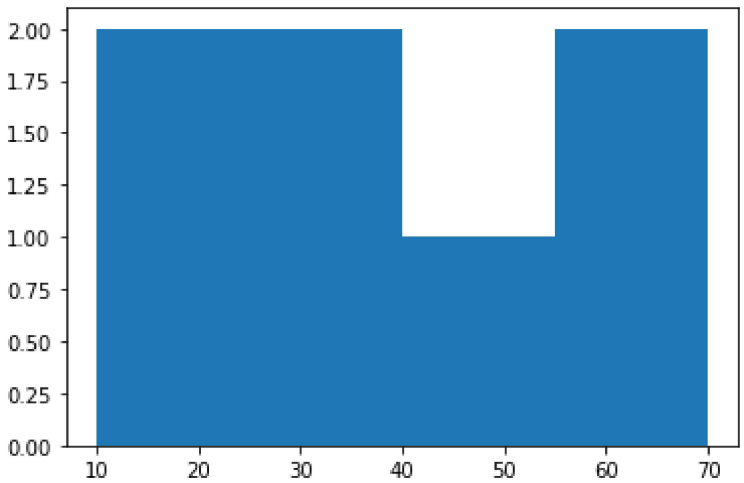


In [22]:

```
#Using Bins

data=[32,45,56,10,15,27,61]
plt.hist(data,bins=[10,25,40,55,70]) # Bin size is large
```

Out[22]:

```
(array([2., 2., 1., 2.]),
 array([10, 25, 40, 55, 70]),
 <BarContainer object of 4 artists>)
```



In [23]:

```
df1= pd.read_csv("vk.csv")
df1
```

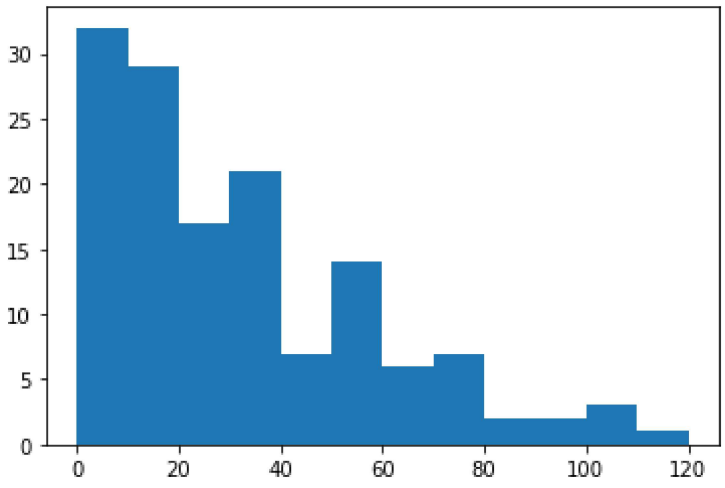
Out[23]:

	match_id	batsman_runs
0	12	62
1	17	28
2	20	64
3	27	0
4	30	10
...	...	...
136	624	75
137	626	113
138	632	54
139	633	0
140	636	54

141 rows × 2 columns

In [25]:

```
plt.hist(df1["batsman_runs"],bins=[0,10,20,30,40,50,60,70,80,90,100,110,120])

plt.show()
```



In [26]:

```
# Handling Bins
Arr= np.load("big-array.npy")
Arr
```

Out[26]:

```
array([33, 39, 37, ..., 33, 30, 39], dtype=int64)
```

In [27]:

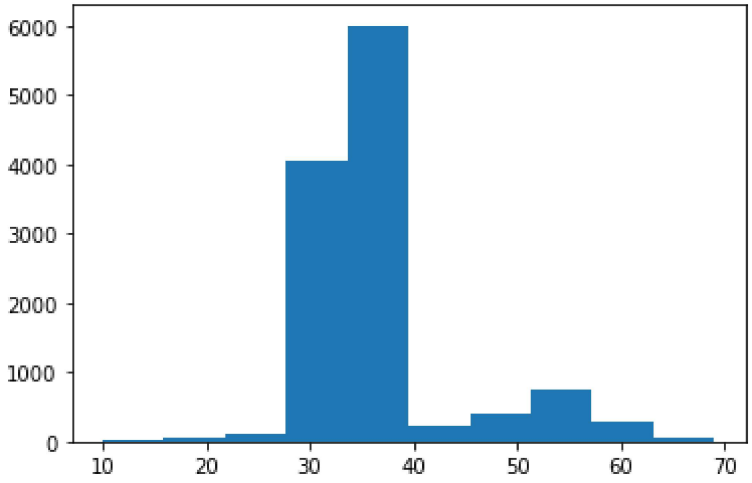
```
Arr.shape
```

Out[27]:

```
(11949,)
```

In [29]:

```
plt.hist(Arr)
plt.show()
```



In [30]:

```
# Some bins contains so many data that some bins are formed properly
```

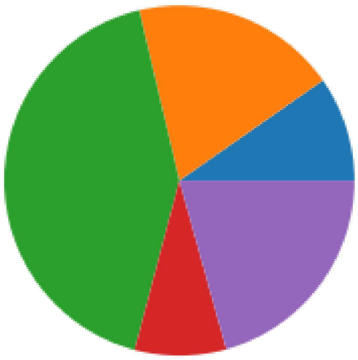
Pie chart

In [31]:

```
# Univariate/Bi-variate
#Categorical Vs Numerical
#Use case- To find the contribution on a standard scale
```

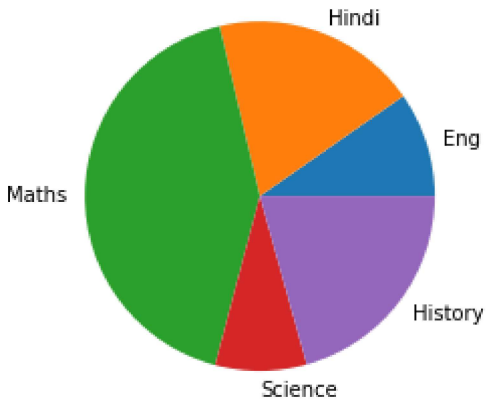
In [33]:

```
#Simple data

df2= [23,45,100,20,49]

plt.pie(df2)
plt.show()
```



In [36]:

```
#Labels

df2= [23,45,100,20,49]
Subjects= ["Eng","Hindi","Maths","Science","History"]

plt.pie(df2,labels=Subjects)
plt.show()
```



In [37]:

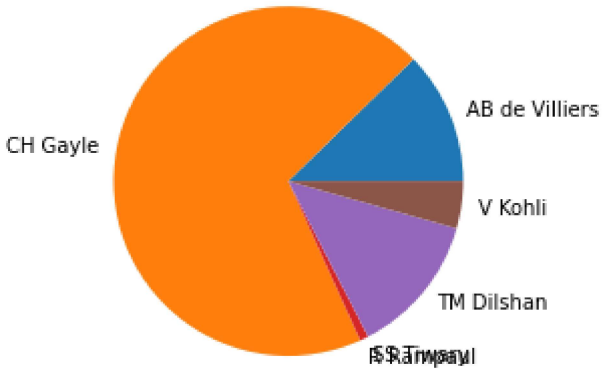
```
#Dataset

df3= pd.read_csv("gayle-175.csv")
df3
```

Out[37]:

	batsman	batsman_runs
0	AB de Villiers	31
1	CH Gayle	175
2	R Rampaul	0
3	SS Tiwary	2
4	TM Dilshan	33
5	V Kohli	11

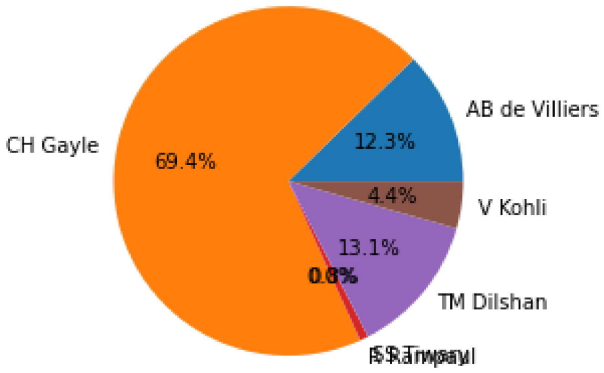
In [40]:

```
plt.pie(df3["batsman_runs"], labels=df3["batsman"])
plt.show()
```



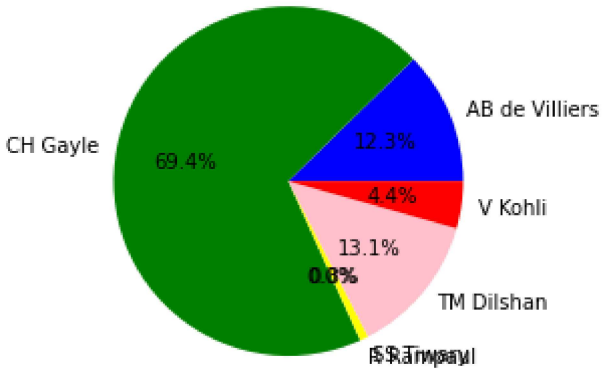
In [42]:

```
# Showing Percentage

plt.pie(df3["batsman_runs"], labels=df3["batsman"], autopct="%0.1f%%")
plt.show()
```



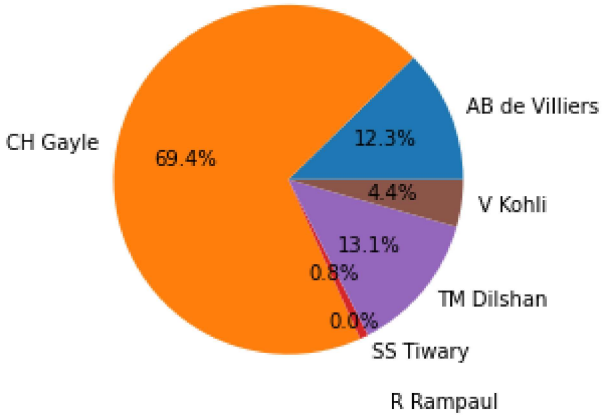
In [44]:

```
plt.pie(df3["batsman_runs"], labels=df3["batsman"], autopct="%0.1f%%", colors=["blue","g
plt.show()
```



In [47]:

```
#Exploding

plt.pie(df3["batsman_runs"], labels=df3["batsman"], autopct="%0.1f%%",explode=[0,0,0.3,0
plt.show()
```



In [ ]:

```

```